

10/591 933
IAP9 Rec'd PCT/PTO 06 SEP 2006

中华人民共和国国家知识产权局
STATE INTELLECTUAL PROPERTY OFFICE
OF THE PEOPLE'S REPUBLIC OF CHINA



证 明
CERTIFICATE

**CERTIFIED COPY OF
PRIORITY DOCUMENT**

本证明之附件是向中国专利局作为受理局提交的下列国际申请副本
THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY OF THE BELOW
IDENTIFIED INTERNATIONAL APPLICATION THAT WAS FILED WITH THE
CHINESE PATENT OFFICE AS RECEIVING OFFICE

申请号: PCT/CN2005/000852

NATIONAL APPLICATION NUMBER

申请日: 15.6月 2005 (15.06.2005)

ATILNAL FILING DATE

名称:

USING TRANSACTED WRITES AND CACHING MECHANISM

OF INVENTION

TO IMPROVE WRITE PERFORMANCE IN MULTI-LEVEL

CELL FLASH MEMOTY

中华人民共和国国家知识产权局局长

COMMISSIONER OF THE STATE INTELLECTUAL PROPERTY
OFFECE OF THE PEOPLE'S REPUBLIC OF CHINA

二零零六年八月十四日

AUGUST 14, 2006

BEST AVAILABLE COPY

HOME COPY

1/4

PCT REQUEST

Original (for SUBMISSION) - printed on 15.06.2005 02:25:19 PM

054429 PC

0	For receiving Office use only	
0-1	International Application No.	PCT/CN 2005 / 0 0 0 8 5 2
0-2	International Filing Date	1 5 . 6 月 2005 (1 5 . 0 6 . 2 0 0 5)
0-3	Name of receiving Office and "PCT International Application"	RO/CN 中华人民共和国国家知识产权局 PCT International Application
0-4	Form - PCT/RO/101 PCT Request	
0-4-1	Prepared using	PCT-EASY Version 2.92 (updated 01.11.2003)
0-5	Petition The undersigned requests that the present international application be processed according to the Patent Cooperation Treaty	
0-6	Receiving Office (specified by the applicant)	China Intellectual Property Office (RO/CN)
0-7	Applicant's or agent's file reference	054429 PC
I	Title of invention	USING TRANSACTED WRITES AND CACHING MECHANISM TO IMPROVE WRITE PERFORMANCE IN MULTI-LEVEL CELL FLASH MEMOTY
II	Applicant	
II-1	This person is:	applicant only
II-2	Applicant for	all designated States except US
II-4	Name	INTEL CORPORATION
II-5	Address:	2200 Mission College Boulevard Santa Clara, CA 95052 United States of America
II-6	State of nationality	US
II-7	State of residence	US
III-1	Applicant and/or inventor	
III-1-1	This person is:	applicant and inventor
III-1-2	Applicant for	US only
III-1-4	Name (LAST, First)	SURESH, Nagarajan
III-1-5	Address:	4320 Grafton Circle, Mather, CA 95655 United States of America
III-1-6	State of nationality	IN
III-1-7	State of residence	US

PCT REQUEST

Original (for SUBMISSION) - printed on 15.06.2005 02:25:19 PM

III-2	Applicant and/or inventor	
III-2-1	This person is:	applicant and inventor
III-2-2	Applicant for	US only
III-2-4	Name (LAST, First)	WANG, Hongyu
III-2-5	Address:	Room 301, No.8, Lane 158 Minhang District Shanghai 201100 China
III-2-6	State of nationality	CN
III-2-7	State of residence	CN
IV-1	Agent or common representative; or address for correspondence The person identified below is hereby/has been appointed to act on behalf of the applicant(s) before the competent International Authorities as:	agent
IV-1-1	Name	SHANGHAI PATENT & TRADEMARK LAW OFFICE, LLC
IV-1-2	Address:	435 Guiping Road Shanghai 200233 China
IV-1-3	Telephone No.	86-21-64853500
IV-1-4	Facsimile No.	86-21-64828651
IV-1-5	e-mail	sptl@siit.intnet.com.cn
V	Designation of States	
V-1	Regional Patent (other kinds of protection or treatment, if any, are specified between parentheses after the designation(s) concerned)	AP: BW GH GM KE LS MW MZ SD SL SZ TZ UG ZM ZW and any other State which is a Contracting State of the Harare Protocol and of the PCT EA: AM AZ BY KG KZ MD RU TJ TM and any other State which is a Contracting State of the Eurasian Patent Convention and of the PCT EP: AT BE BG CH&LI CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL PT RO SE SI SK TR and any other State which is a Contracting State of the European Patent Convention and of the PCT OA: BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG and any other State which is a member State of OAPI and a Contracting State of the PCT

PCT REQUEST

054429 PC

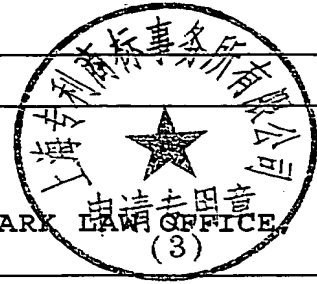
Original (for SUBMISSION) - printed on 15.06.2005 02:25:19 PM

V-2	National Patent (other kinds of protection or treatment, if any, are specified between parentheses after the designation(s) concerned)	AE AG AL AM AT AU AZ BA BB BG BR BW BY BZ CA CH&LI CN CO CR CU CZ DE DK DM DZ EC EE EG ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NA NI NO NZ OM PG PH PL PT RO RU SC SD SE SG SK SL SY TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW	
V-5	Precautionary Designation Statement In addition to the designations made under items V-1, V-2 and V-3, the applicant also makes under Rule 4.9(b) all designations which would be permitted under the PCT except any designation(s) of the State(s) indicated under item V-6 below. The applicant declares that those additional designations are subject to confirmation and that any designation which is not confirmed before the expiration of 15 months from the priority date is to be regarded as withdrawn by the applicant at the expiration of that time limit.		
V-6	Exclusion(s) from precautionary designations	NONE	
VI	Priority claim	NONE	
VII-1	International Searching Authority Chosen	China Intellectual Property Office (ISA/CN)	
VIII	Declarations	Number of declarations	
VIII-1	Declaration as to the identity of the inventor	-	
VIII-2	Declaration as to the applicant's entitlement, as at the international filing date, to apply for and be granted a patent	-	
VIII-3	Declaration as to the applicant's entitlement, as at the international filing date, to claim the priority of the earlier application	-	
VIII-4	Declaration of inventorship (only for the purposes of the designation of the United States of America)	-	
VIII-5	Declaration as to non-prejudicial disclosures or exceptions to lack of novelty	-	
IX	Check list	number of sheets	electronic file(s) attached
IX-1	Request (including declaration sheets)	4	-
IX-2	Description	17	-
IX-3	Claims	6	-
IX-4	Abstract	1	EZABST00.TXT
IX-5	Drawings	5	-
IX-7	TOTAL	33	
	Accompanying items	paper document(s) attached	electronic file(s) attached
IX-8	Fee calculation sheet	✓	-
IX-17	PCT-EASY diskette	-	Diskette
IX-19	Figure of the drawings which should accompany the abstract		

PCT REQUEST

Original (for SUBMISSION) - printed on 15.06.2005 02:25:19 PM

IX-20	Language of filing of the international application	English
X-1	Signature of applicant, agent or common representative	
X-1-1	Name	SHANGHAI PATENT & TRADEMARK LAW OFFICE, LLC



FOR RECEIVING OFFICE USE ONLY

10-1	Date of actual receipt of the purported international application	15.6月2005 (15.06.2005)
10-2	Drawings:	
10-2-1	Received	
10-2-2	Not received	
10-3	Corrected date of actual receipt due to later but timely received papers or drawings completing the purported international application	
10-4	Date of timely receipt of the required corrections under PCT Article 11(2)	
10-5	International Searching Authority	ISA/CN
10-6	Transmittal of search copy delayed until search fee is paid	

FOR INTERNATIONAL BUREAU USE ONLY

11-1	Date of receipt of the record copy by the International Bureau	
------	--	--

PCT (ANNEX - FEE CALCULATION SHEET)

054429 PC

Original (for SUBMISSION) - printed on 15.06.2005 02:25:19 PM

(This sheet is not part of and does not count as a sheet of the international application)

0	For receiving Office use only																																																							
0-1	International Application No.	PCT/CN 2005 / 0 0 0 8 5 2																																																						
0-2	Date stamp of the receiving Office	1 5 · 6月 2005 (1 5 · 0 6 · 2 0 0 5)																																																						
0-4	Form - PCT/RO/101 (Annex)																																																							
0-4-1	PCT Fee Calculation Sheet Prepared using	PCT-EASY Version 2.92 (updated 01.11.2003)																																																						
0-9	Applicant's or agent's file reference	054429 PC																																																						
2	Applicant	INTEL CORPORATION, et al.																																																						
12	Calculation of prescribed fees	<table border="1"> <thead> <tr> <th>fee amount/multiplier</th> <th>Total amount (Equivalent in local currency of CHF)</th> <th>Total amounts (CNY)</th> </tr> </thead> <tbody> <tr> <td>12-1 Transmittal fee T ⇒</td> <td></td> <td>500</td> </tr> <tr> <td>12-2-1 Search fee S ⇒</td> <td></td> <td>1,500</td> </tr> <tr> <td>12-2-2 International search to be carried out by CN</td> <td></td> <td></td> </tr> <tr> <td>12-3 International fee</td> <td></td> <td></td> </tr> <tr> <td>Basic fee (first 30 sheets) b1 1,400 CHF</td> <td></td> <td></td> </tr> <tr> <td>12-4 Remaining sheets 3</td> <td></td> <td></td> </tr> <tr> <td>12-5 Additional amount (X) 15 CHF</td> <td></td> <td></td> </tr> <tr> <td>12-6 Total additional amount b2 45 CHF</td> <td></td> <td></td> </tr> <tr> <td>12-7 b1 + b2 = B 1,445 CHF</td> <td></td> <td></td> </tr> <tr> <td>12-8 Designation fees</td> <td></td> <td></td> </tr> <tr> <td>Number of designations contained in international application 100</td> <td></td> <td></td> </tr> <tr> <td>12-9 Number of designation fees payable (maximum 5) 5</td> <td></td> <td></td> </tr> <tr> <td>12-10 Amount of designation fee (X) 0 CHF</td> <td></td> <td></td> </tr> <tr> <td>12-11 Total designation fees D 0 CHF</td> <td></td> <td></td> </tr> <tr> <td>12-12 PCT-EASY fee reduction R -100 CHF</td> <td></td> <td></td> </tr> <tr> <td>12-13 Total international fee (B+D-R) I ⇒</td> <td>1,345</td> <td></td> </tr> <tr> <td>12-17 TOTAL FEES PAYABLE (T+S+I+P) ⇒</td> <td>1,345</td> <td>2,000</td> </tr> </tbody> </table>	fee amount/multiplier	Total amount (Equivalent in local currency of CHF)	Total amounts (CNY)	12-1 Transmittal fee T ⇒		500	12-2-1 Search fee S ⇒		1,500	12-2-2 International search to be carried out by CN			12-3 International fee			Basic fee (first 30 sheets) b1 1,400 CHF			12-4 Remaining sheets 3			12-5 Additional amount (X) 15 CHF			12-6 Total additional amount b2 45 CHF			12-7 b1 + b2 = B 1,445 CHF			12-8 Designation fees			Number of designations contained in international application 100			12-9 Number of designation fees payable (maximum 5) 5			12-10 Amount of designation fee (X) 0 CHF			12-11 Total designation fees D 0 CHF			12-12 PCT-EASY fee reduction R -100 CHF			12-13 Total international fee (B+D-R) I ⇒	1,345		12-17 TOTAL FEES PAYABLE (T+S+I+P) ⇒	1,345	2,000
fee amount/multiplier	Total amount (Equivalent in local currency of CHF)	Total amounts (CNY)																																																						
12-1 Transmittal fee T ⇒		500																																																						
12-2-1 Search fee S ⇒		1,500																																																						
12-2-2 International search to be carried out by CN																																																								
12-3 International fee																																																								
Basic fee (first 30 sheets) b1 1,400 CHF																																																								
12-4 Remaining sheets 3																																																								
12-5 Additional amount (X) 15 CHF																																																								
12-6 Total additional amount b2 45 CHF																																																								
12-7 b1 + b2 = B 1,445 CHF																																																								
12-8 Designation fees																																																								
Number of designations contained in international application 100																																																								
12-9 Number of designation fees payable (maximum 5) 5																																																								
12-10 Amount of designation fee (X) 0 CHF																																																								
12-11 Total designation fees D 0 CHF																																																								
12-12 PCT-EASY fee reduction R -100 CHF																																																								
12-13 Total international fee (B+D-R) I ⇒	1,345																																																							
12-17 TOTAL FEES PAYABLE (T+S+I+P) ⇒	1,345	2,000																																																						
12-19	Mode of payment	authorization to charge deposit account																																																						
12-20	Deposit account instructions The receiving Office:	China Intellectual Property Office (RO/CN)																																																						
12-20-1	Authorization to charge the total fees indicated above.	✓																																																						
12-20-2	Authorization to charge any deficiency or credit any overpayment in the total fees indicated above.	✓																																																						
12-20-3	Authorization to charge the fee for priority document.	✓																																																						
12-21	Deposit account No.	SPTL																																																						
12-22	Date	15 June 2005 (15.06.2005)																																																						

PCT (ANNEX - FEE CALCULATION SHEET)

Original (for SUBMISSION) - printed on 15.06.2005 02:25:19 PM

054429 PC

12-23	Name and signature	SHANGHAI PATENT & TRADEMARK LAW OFFICE LLC
-------	--------------------	---



VALIDATION LOG AND REMARKS

13-2-1	Validation messages Request	Green? The title of the invention shall be short and precise. Please verify.
13-2-3	Validation messages Names	Green? Applicant 1.:Telephone No. missing
		Green? Applicant 1.:Facsimile No. missing
13-2-4	Validation messages Priority	Green? No priority of an earlier application has been claimed. Please verify
13-2-7	Validation messages Contents	Yellow! The power of attorney or a copy of the general power of attorney will need to be furnished unless all applicants sign the request form.
		Green? Figure of the drawings which should accompany the abstract not specified. Please verify.
13-2-8	Validation messages Fees	Green? Please confirm that fee schedule utilized is the latest available
		Yellow Fee amount(s) should not equal zero.
13-2-9	Validation messages Payment	Green? Please ensure that you have a valid deposit account with the receiving Office selected.
13-2-10	Validation messages Annotate	Green? The name of the person signing the request or/and the capacity in which the person signs has/have not been indicated. Please be informed that some receiving Offices require that this information be present along with the signature.
13-2-11	Validation messages For receiving Office/International Bureau use only	Green? Verify electronic data for consistency against printed form.

USING TRANSACTED WRITES AND CACHING MECHANISM TO IMPROVE WRITE PERFORMANCE IN MULTI-LEVEL CELL FLASH MEMORY

BACKGROUND

[0001] Non-volatile memory products for electronic equipments, such as cell phones, digital cameras, computers, etc., are widely used today. When writing a file to the non-volatile memory, data fragments, sequence tables and their associated headers may be written to the non-volatile memory one by one. Here, the data fragments store user data of the file and the sequence tables may comprises sequence table entries to store memory locations of the data fragments. A sequence table entry may comprise a data field for a memory location, and, an entry allocating field and entry valid field to protect the sequence table entry in case of power loss.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The invention described herein is illustrated by way of example and not by way of limitation in the accompanying figures. For simplicity and clarity of illustration, elements illustrated in the figures are not necessarily drawn to scale. For example, the dimensions of some elements may be exaggerated relative to other elements for clarity. Further, where considered appropriate, reference labels have been repeated among the figures to indicate corresponding or analogous elements.

[0003] Fig.1 illustrates a memory writing device;

[0004] Fig. 2 illustrates an embodiment of a file system maintained in a non-volatile memory of the memory writing device of Fig. 1;



- [0005] Fig. 3 illustrates an embodiment of a sequence table temporarily stored in a volatile memory of the memory writing device of Fig. 1;
- [0006] Fig. 4a illustrates a transaction indicator in the file system of Fig. 2;
- [0007] Fig. 4b illustrates a sequence table entry in the file system of Fig. 2;
- [0008] Fig. 4c illustrates a sequence table header in the file system of Fig. 2;
- [0009] Fig. 4d illustrate a data fragment header in the file system of Fig. 2;
- [0010] Fig.5 illustrates an embodiment of a memory writing method that may be used by the device of Fig. 1 ;
- [0011] Fig. 6 illustrates a comparison between two memory writing methods.

DETAILED DESCRIPTION

- [0012] The following description describes techniques for memory writing method and system. In the following description, numerous specific details such as logic implementations, pseudo-code, means to specify operands, resource partitioning/sharing/duplication implementations, types and interrelationships of system components, and logic partitioning/integration choices are set forth in order to provide a more thorough understanding of the present invention. However, the invention may be practiced without such specific details. In other instances, control structures, gate level circuits and full software instruction sequences have not been shown in detail in order not to obscure the invention. Those of ordinary skill in the art, with the included descriptions, will be able to implement appropriate functionality without undue experimentation.
- [0013] References in the specification to “one embodiment”, “an embodiment”, “an example embodiment”, etc., indicate that the embodiment described may

4/2

include a particular feature, structure, or characteristic, but every embodiment may not necessarily include the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to effect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described.

[0014] Embodiments of the invention may be implemented in hardware, firmware, software, or any combination thereof. Embodiments of the invention may also be implemented as instructions stored on a machine-readable medium, which may be read and executed by one or more processors. A machine-readable medium may include any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computing device). For example, a machine-readable medium may include read only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; and others.

[0015] Fig. 1 shows a memory writing device 100 according to one embodiment. The device 100 includes a processor 101, a non-volatile memory 102, and a volatile memory 103. The processor 101 may be any type of processor adapted to perform operations to the non-volatile memory 102. For example, the processor 101 may be a microprocessor, a digital signal processor, a microcontroller, or the like. The non-volatile memory 102 may comprises a flash memory, such as NOR flash memory, NAND flash memory. A volatile memory 103 may comprises various types of random access memories (RAMs), for

3/2

example, dynamic random access memory (DRAM), synchronous direct random access memory (SDRAM), double data rate (DDR) SDRAMs, or other memories.

[0016] The processor 101, the non-volatile memory 102 and the volatile memory 103 may be coupled by buses 104-105. In some embodiments, the processor 101, the non-volatile memory 102 and the volatile memory 103 may be included on an integrated circuit board, and the buses 104-105 may be implemented using traces on the circuit board.

[0017] According to various embodiments, the processor 101 may perform operations to non-volatile memory 102 and volatile memory 103. In one embodiment, the processor 101 may control writing a partial file or a whole file to the non-volatile memory 102. The file may comprise a plurality of data fragments, at least one sequence table including a plurality of sequence table entries to identify memory locations of the data fragments, and other associated information. Before being written to the non-volatile memory 102, the sequence table entries may be temporarily stored to the volatile memory 103 and then may be transferred from the volatile memory 103 to the non-volatile memory 102 under the control of the processor 101.

[0018] Referring now to Figs. 2-3 and Figs. 4a-4d, a file system of the non-volatile memory 102 and a sequence table temporarily stored in the volatile memory 103 will be described in detail. Figs. 2 and 3 respectively show the file system stored in the non-volatile memory 102 and the sequence table temporarily stored in the volatile memory 103. Figs. 4a-4d respectively show a transaction indicator (TI), sequence table entry (STE), sequence table header (STH) and data fragment header (DFH) of the file system shown in Fig. 2.

40

[0019] According to various embodiments, the file system 200 maintained in the non-volatile memory may hold files written to the non-volatile memory and any associated information about the files. The file system 200 may include a file information structure 210 and a number of storage blocks 220, 250, 260 and 280, etc. For NOR flash memory manufactured by Intel Corporate, Santa Clara, California, a storage block may comprise 64k or 128k bytes. For NAND flash memory manufactured by SAMSUNG, Korea, a block may comprise 16k or 32k bytes. As shown in Fig. 2, the file information structure 210 may include a transaction indicator (TI) 215 to indicate states of a transaction for writing a file to the non-volatile memory 102, including a begin state and end state of the transaction. The file information structure 210 may further include a number of sequence table pointers (STP) (e.g., STPs 211, 212 and 213) to respectively point to sequence tables (ST) in the storage blocks of the non-volatile memory 102 (e.g., STs 221, 252 and 253 in the storage blocks 220 and 250). The file information structure 210 may also include the filename, creation date, size of a file, or any other information relating to a file. Storage blocks 220, 250, 260 and 280 may represent blocks of memory of the non-volatile memory 102. In some embodiments, each storage block may be divided into a number of fragments to hold a portion of a file. For example, as shown in Fig. 2, storage blocks 220 and 250 may include fragments to hold the sequence tables (e.g., STs 221, 252 and 253) of the file, and storage blocks 260 and 280 may include fragments to hold user data (e.g., data fragments 261, 262 and 283) of the file. For NOR flash memory manufactured by Intel Corporate, Santa Clara, California, US, a



fragment may comprise 512 or 1k bytes. It shall be understood that storage blocks may be used to hold other data than those of the file.

[0020] Moreover, each sequence table may have a sequence table header (STH) associated therewith, and each data fragment may have a data fragment header (DFH) associated therewith. For example, as shown in Fig. 2, the sequence tables (ST) 221, 252 and 253 have their individually associated headers, i.e., the sequence table headers (STH) 221', 252' and 253'. Similarly, the data fragments (DF) 261, 262 and 283 have their individually associated headers, i.e., the data fragment headers (DFH) 261', 262' and 283'. Detailed structure of the sequence table (ST) 221 is shown in Fig. 2. For simplicity, detailed structures of the sequence tables (ST) 252 and 253 which are the same as that of the sequence table (ST) 221 are omitted in Fig. 2. As depicted, the sequence table 221 may include a number of sequence table entries (STE), for example, sequence table entries (STE) 2211, 2212 and 2213, to store memory locations of corresponding data fragments (DF) 261, 262 and 283. In one embodiment, a sequence table may be the same size as a fragment which may be less than the size of a storage block. Accordingly, a single sequence table may suffice for a file with a small size; however, a file with a relatively large size may include more than one sequence table, such as sequence tables (ST) 221, 252 and 253.

[0021] It shall be appreciated that a file in the file system 200 may include any number of storage blocks to hold any number of data fragments and sequence tables. A sequence table may include any number of sequence table entries to point to the data fragments, as long as within its size limitation. The file may



further include any other kinds of data structures than those shown in Fig. 2. In another embodiment, the file information structure 210 of the file system 200 may include a pointer to point to a storage block in the file system 200 which may be used to hold sequence table pointers to point to the sequence tables. The file information structure 210 may further include different transaction indicators for different files in the file system 200. Other embodiments may implement other modifications and variations to the structure of the file system 200.

[0022] Detailed structure of the transaction indicator (TI) 215 is shown in Fig. 4a.

As depicted, the transaction indicator 215 may include two transaction state fields: a transaction begin field, indicating the begin state of the transaction for writing a file; and a transaction end field, indicating the end state of the transaction. In an embodiment, the transaction begin field may comprises 2 bits to represent the transaction begin state, and the transaction end field may comprises 2 bits to represent the transaction end state.

[0023] In an embodiment, the transaction begin field and transaction end field of the transaction indicator (TI) 215 may each include 1 bit. In another embodiment, the transaction begin field may have 2 bits while the transaction end field may have 1 bit. Other embodiments may utilize transaction indicators (TI) 215 having a different structure and/or a different number of bits.

[0024] The transaction indicator may support a power loss recovery feature to keep data consistency of the file to be written in the non-volatile memory 102. Particularly, when power loss happens after the transaction for writing the file is started, but before the transaction ends, parts of the file which have been written



to the non-volatile memory may be deleted automatically by the power loss recovery feature.

[0025] Referring now to Fig. 4b, the detailed structure of the sequence table entry 221 is shown. In Fig. 4b, the sequence table entry may include a data entry field to identify a memory location of a data fragment. Before a sequence table comprising a number of the sequence table entries is written to the non-volatile memory, the sequence table may be temporarily stored in the volatile memory 103. Fig. 3 shows the data structure of a sequence table temporarily stored in the volatile memory 103. It can be seen that the sequence table stored in volatile memory 103 has the same structure as that in the non-volatile memory 102.

[0026] Each sequence table has a sequence table header (STH) associated therewith. Detailed data structure of the sequence table header 221' is shown in Fig. 4c. In Fig. 4c, the sequence table header has two fields representing two sequence table header states: a sequence table header allocating field (st_hdr_allocating field) to represent 'sequence table header allocating' state; a sequence table header valid field (st_hdr_valid field) to represent 'sequence table header valid' state. Data structures of the sequence table headers 252' and 253' are the same as that of the sequence table header 221'.

[0027] Each data fragment may have a data fragment header (DFH) associated therewith. Detailed data structure of the data fragment header 261' is shown in Fig. 4d. As depicted, the data fragment header may include two fields representing two data fragment header states: a data fragment header allocating field (df_hdr_allocating field) to represent 'data fragment header allocating' state; a data fragment header valid field (df_hdr_valid field) to represent 'data fragment



header valid' state. The data structures of the data fragment headers (DFH) 262' and 283' may be implemented in a manner similar to the data fragment header (DFH) 261'.

[0028] It shall be appreciated that the sequence table header and/or data fragment header may further include other data structures, such as a header invalid field (hdr_invalid field) to represent 'header invalid' state. The sequence table header and data fragment header may support a power loss recovery feature for their associated sequence table and data fragment in the non-volatile memory 102. Namely, if power loss happens before validation of a header, its associated sequence table or data fragment may be deleted automatically.

[0029] Now referring to Fig. 5, a method of writing a file to the non-volatile memory according to an embodiment will be described in detail below, the file having the data structure as shown in Fig. 2. The method may be implemented by the device 100 as shown in Fig. 1.

[0030] In block 501, the processor 101 may start a transaction for writing a file by changing state of the transaction indicator (TI) 215 in the non-volatile memory 102 to a transaction begin state. For implementation, in an embodiment where the transaction indicator 215 includes a transaction begin field having 2 bits to identify the transaction begin state, the processor 101 may write to the transaction begin field, thereby changing the 2 state bits in the transaction begin field from "11" to "00" to indicate begin of the transaction for writing the file. This change from "11" to "00" may be accomplished using bit twiddling write mode which is a special word programming write mode supported by some non-volatile memory devices. Namely, only two bits of a word are programmed in the bit

twiddling mode. However, other embodiments may utilize a different data structure of the transaction indicator and/or a different write mode.

[0031] In block 502, the processor 101 may control changing state of a data fragment header (e.g., DFH 261' in Fig. 2) to a 'data fragment header allocating' state which indicates a fragment of a block in the non-volatile memory 102 is allocated to hold a data fragment of the file. Again, this change in state may be accomplished by writing to a data fragment header allocating field (df_hdr_allocating field) of the data fragment header in bit twiddling write mode.

[0032] In block 503, the processor 101 may control writing to the non-volatile memory 102 a data fragment associated with the data fragment header in block 502. For example, as shown in Fig. 2, data fragment (DF) 261 associated with data fragment header (DFH) 261' in block 502 may be written to the non-volatile memory in block 503. This writing of the data fragment may be accomplished in one embodiment with buffer programming write mode to write user data of the file.

[0033] In block 504, the processor 101 may control writing a sequence table entry to the volatile memory 103. The sequence table entry may comprise a data entry field to identify the memory location of the corresponding data fragment written in block 503. For example, the sequence table entry (STE) 2211 as shown in Fig. 3, which identifies the memory location of the corresponding data fragment (DF) 261 written in block 503, may be written to the volatile memory 103 in block 504. Since the sequence table entry may be written to a volatile memory, the sequence table entry may not need to add power loss recovery information for each sequence table entry. That is to say, the sequence table entry may not

comprise fields to identify entry states, such as entry allocating state and entry valid state, for the purpose of power loss recovery.

[0034] In block 505, the processor 101 may determine whether all of the data fragments which carry user data of the file are written to the non-volatile memory. Namely, the processor 101 may determine whether writing the user data of the file to the non-volatile memory is completed. If not, the processor 101 may continue to block 506 where the processor 101 may determine whether the sequence table temporarily stored in the volatile memory 103 (e.g., sequence table 221 in Fig. 3) is full. Since a sequence table may have a limited size, a file with a large size may need more than one sequence table to accommodate a large amount of sequence table entries to identify the memory locations for all of the data fragments of the file. If the processor 101 determines that the sequence table temporarily stored in the volatile memory 103 is not full, the processor 101 may return to block 502 to allocate another data fragment. Namely, the processor 101 may continue to write data fragments to the non-volatile memory 102 and to write their corresponding sequence table entries to the volatile memory 103.

[0035] If the sequence table stored in the volatile memory 103 is determined to be full in block 506, the processor 101 may continue to block 507 to allocate another sequence table header. Namely, the processor 101 in block 507 may change state of a sequence table header to 'sequence table header allocating' state in the non-volatile memory 102, indicating that a fragment of a block in the non-volatile memory is allocated to hold a sequence table which may be written to the non-volatile memory. For example, the sequence table 221 temporarily stored in the volatile memory 103 (STH) 221' in Fig. 2 may be changed to a

3/12

'sequence table header allocating' state. This change in state may be implemented in one embodiment by writing a sequence table header allocating field (st_hdr_allocating field) of the sequence table header (STH) 221' in bit twiddling write mode.

[0036] Then, in block 508, the processor 101 may control writing of the sequence table temporarily stored in the volatile memory 103 to the non-volatile memory 102. For example, the sequence table (ST) 221 in Fig. 3 may be written from the volatile memory to the non-volatile memory in block 508. This writing of the sequence table may be implemented in one embodiment with buffer programming write mode. By completion of block 508, a part of the file corresponding to one sequence table is written to the non-volatile memory.

[0037] Then, the processor 101 may return to block 502 to continue writing other parts of the file corresponding to other sequence tables such as, for example, the sequence tables (ST) 252 and 253 as shown in Fig. 2. However, modifications and variations to the above are possible. For instance, when a size limitation for a sequence table is set to be large enough to accommodate locations of data fragments of the file, blocks 506-508 may be omitted from Fig. 5.

[0038] If, in block 505, the processor 101 determines that writing the user data of the file to the non-volatile memory is completed, the processor may continue to block 509. Namely, the processor 101 may control changing state of a sequence table header in the non-volatile memory 102 to a sequence table header allocating state, which indicates a fragment of a block in the non-volatile memory is allocated to hold a sequence table which may be written to the non-volatile memory. In an embodiment that may utilize more than one sequence table



for the file, the fragment allocated in block 509 may be used to hold the last sequence table of the file. For example, a file may comprise three sequence tables (ST) 221, 252 and 253 as shown in Fig. 2. After parts of the file corresponding to the sequence tables 212 and 252 are written to the non-volatile memory 102 and processor 101 is in the process of writing the last part of the file corresponding to the last sequence table 253, the processor 101 may determine that all of the data fragment carrying user data of the file have been written to the non-volatile memory. The processor 101 may change the state of a sequence table header associated with the last sequence table, e.g., sequence table header (STH) 253' to sequence table header allocating state which indicates a fragment of a storage block is allocated to hold the sequence table 253 in the non-volatile memory 102. This allocating of the sequence table header may be implemented by one embodiment by writing a sequence table header allocating field of the sequence table header (STH) 253' in bit twiddling write mode.

[0039] Then, in block 510, the processor 101 may control writing of the sequence table temporarily stored in the volatile memory 103 to the non-volatile memory 102. In an embodiment, the last sequence table 253 which is temporarily stored in the volatile memory 103 may be written to the non-volatile memory 102 in block 510.

[0040] Then, in block 511, the processor 101 may control changing the state of the transaction indicator 215 to a transaction end state. For implementation, in an embodiment where the transaction indicator 215 includes a transaction end field having 2 bits to identify the transaction end state as shown in Fig. 4a, the processor 101 may write to the transaction end field, thereby changing the 2



state bits in the transaction end field from "11" to "00" to indicate end of writing the file. The following table 1 shows how states of the transaction corresponds to the pair of the transaction begin and transaction end fields according with an embodiment.

Transaction Begin Field	Transaction End Field	Transaction State
00	11	begin
00	00	end

Table 1

[0041] This ending of the transaction may be implemented by an embodiment via a bit twiddling write mode. However, other embodiments may utilize a transaction indicator with a different structure, a different number of bits, and/or a different write mode.

[0042] Then, in block 512, the processor 101 may change the state of data fragment headers associated with the data fragments of the file which have been written to the non-volatile memory from header allocating state to a header valid state. Completion of block 512 indicates that the fragments of the blocks, which have been allocated in block 502 to hold the data fragments of the file, are validated. This validation may be implemented by an embodiment by writing header valid fields of the data fragment headers associated with the data fragments, which have been written to the non-volatile memory, in bit twiddling write mode.

[0043] Then, in block 513, the processor 101 may control changing state of the sequence table header(s) associated with the sequence table(s) of the file from the sequence table header allocating state to the sequence table header valid



state. In an embodiment that the file has only one sequence table, its associated sequence table header is validated in block 513. In another embodiment that the file has more than one sequence tables, their associated sequence table headers are validated in block 513. Completion of block 513 indicates that the fragment(s) of storage block(s), which had been allocated in blocks 507 and/or 509 to hold the sequence table(s) of the file, is validated. This validation may be implemented in one embodiment by writing sequence table header valid field(s) of the sequence table header(s) associated with the sequence table(s) in bit twiddling write mode.

[0044] It can be seen from the flow chart of Fig. 5 that the data fragment headers, sequence table header(s) and the transaction indicator may support a power loss recovery feature. Namely, if the power loss occurs when the transaction of writing the file begins but not ends, which corresponding to the state of the transaction indicator is changed to the transaction begin state, the data fragments and sequence table(s) of the file, whose associated headers have not been validated yet, may be deleted from the non-volatile memory 102.

[0045] If the power loss occurs when the transaction of writing the file ends, which corresponding to the state of the transaction indicator is changed to the transaction end state, the data fragments and sequence table(s) of the file may be maintained in the non-volatile memory 102 by validating their associated headers. By this way, data consistency for the whole file may be maintained.

[0046] However, other embodiments may utilize modifications and variations of the above described power loss recovery feature. For instance, when data consistency is desired on sequence table level but not on file level, the



transaction indicator may be omitted. In such case, after each part of a file corresponding to each sequence table is written to the non-volatile memory, including writing data fragments of that part to the non-volatile memory and writing the corresponding sequence table from the volatile memory to the non-volatile memory, each of the associated sequence table header and data fragment headers may be validated.

[0047] Pseudo code of power loss recovery initialization in the non-volatile memory according to an embodiment may be:

```

If (transaction begin but not end) {
    Scan the NV-memory to invalidate all new data
}
Else {
    Go through sequence table(s) to validate all new
    data
}

```

[0048] It shall be well understood by a skilled person in the art that the above-described method may be applied to write a part of a file to the non-volatile memory, but not whole of the file. Then, other parts of the file may be written by using other memory writing methods. In such case, the transaction indicator may indicate begin and end states of a transaction for writing a part of a file to the non-volatile memory. Then, a sequence table of the part of the file may be transferred from the volatile memory to the non-volatile memory if the sequence table is full or all of data fragments of that part of the file have been written to the non-volatile memory.

[0049] Fig. 6 illustrates the comparison between two memory writing methods wherein a file to be written comprises 64 data fragments and the sequence tables have a sequence table size limit of 64 entries. The upper part of Fig.5 shows the

342

steps taken to write the file of 64 data fragments according to a first writing method, while the lower part of Fig. 5 shows those according to a second writing method.

[0050] It can be seen from Fig. 6 that according to the first writing method, in order to write the file of 64 data fragments, the device 100 as shown in Fig. 1 may perform 256 bit twiddling, 128 word programming and 64k bytes buffer programming operations, in which the bit twiddling and word programming operations consumes a relatively large amount of time. However, according to the second memory writing method, in order to write the file with the same size, the device 100 as shown in Fig. 1 may only perform 132 bit twiddling, 0 word programming and 64.5k bytes buffer programming operations, which may significantly save file writing time. It shall be understood that the device 100 may be implemented to utilize both the first file writing method of the second writing method. It shall also be understood that either memory writing method may be applied to write a partial file instead of a whole of the file.

[0051] Although the present invention has been described in conjunction with certain embodiments, it shall be understood that modifications and variations may be resorted to without departing from the spirit and scope of the invention as those skilled in the art readily understand. Such modifications and variations are considered to be within the scope of the invention and the appended claims.



What is claimed is:

1. A method, comprising:
writing data fragments to a non-volatile memory; and
updating sequence table entries to a volatile memory that identify
locations of the data fragments written to the non-volatile memory.
2. A method of claim 1, further comprising:
writing the sequence table entries to the non-volatile memory.
3. A method of claim 1, further comprising:
writing the sequence table having the sequence table entries to the
non-volatile memory if the sequence table is full.
4. A memory of claim 1, further comprising:
writing the sequence table having the sequence table entries to the
non-volatile memory if writing the data fragments to the non-volatile memory is
completed.
5. A method of claim 1, further comprising:
updating a transaction indicator in the non-volatile memory prior to writing
a transaction to the non-volatile memory; and
updating the transaction indicator in the non-volatile memory after writing
the transaction to the non-volatile memory,



wherein the transaction comprises the data fragments and the sequence table entries.

6. A method of claims 1, further comprising:

allocating a data fragment header associated to the data fragment in the non-volatile memory prior to writing the data fragment to the non-volatile memory; and

validating the data fragment header after writing the sequence table entries to the non-volatile memory.

7. A method of claim 1, comprising:

allocating a sequence table header associated with the sequence table in the non-volatile memory prior to writing the sequence table to the non-volatile memory; and

validating the sequence table header after writing the sequence table entries to the non-volatile memory.

8. A system, comprising:

a non-volatile memory;

a volatile memory; and

a processor to:

write data fragments to a non-volatile memory; and

update sequence table entries to a volatile memory that identify locations of the data fragments written to the non-volatile memory.

3/12

9. A system of claim 8, wherein the processor further writes the sequence table entries to the non-volatile memory.

10. A system of claim 8, wherein the processor further writes the sequence table having the sequence table entries to the non-volatile memory if the sequence table is full.

11. A system of claim 8, wherein the processor further writes the sequence table having the sequence table entries to the non-volatile memory if writing the data fragments to the non-volatile memory is completed.

12. A system of claim 8, wherein the processor further:
updates a transaction indicator in the non-volatile memory prior to writing a transaction to the non-volatile memory; and
updates the transaction indicator in the non-volatile memory after writing the transaction to the non-volatile memory,
wherein the transaction comprises the data fragments and the sequence table entries.

13. A system of claims 8, wherein the processor further:
allocates a data fragment header associated to the data fragment in the non-volatile memory prior to writing the data fragment to the non-volatile memory;
and



validates the data fragment header after writing the sequence table entries to the non-volatile memory.

14. A system of claim 8, wherein the processor further:

allocates a sequence table header associated with a sequence table in the non-volatile memory prior to writing the sequence table to the non-volatile memory; and

validates the sequence table header after writing the sequence table entries to the non-volatile memory.

15. A machine-readable medium comprising a plurality of instructions which when executed result in an apparatus:

writing data fragments to a non-volatile memory; and

updating sequence table entries to a volatile memory that identify locations of the data fragments written to the non-volatile memory.

16. The machine-readable medium of claim 15 wherein the plurality of instructions further result in the apparatus:

writing the sequence table entries to the non-volatile memory.

17. The machine-readable medium of claim 15 wherein the plurality of instructions further result in the apparatus:

writing the sequence table having the sequence table entries to the non-volatile memory if the sequence table is full.

18. The machine-readable medium of claim 15 wherein the plurality of instructions further result in the apparatus:

writing the sequence table having the sequence table entries to the non-volatile memory if writing the data fragments to the non-volatile memory is completed.

19. The machine-readable medium of claim 15 wherein the plurality of instructions further result in the apparatus:

updating a transaction indicator in the non-volatile memory prior to writing a transaction to the non-volatile memory; and

updating the transaction indicator in the non-volatile memory after writing the transaction to the non-volatile memory,

wherein the transaction comprises the data fragments and the sequence table entries.

20. The machine-readable medium of claim 15 wherein the plurality of instructions further result in the apparatus:

allocating a data fragment header associated to the data fragment in the non-volatile memory prior to writing the data fragment to the non-volatile memory; and

validating the data fragment header after writing the sequence table entries to the non-volatile memory.



21. The machine-readable medium of claim 15 wherein the plurality of instructions further result in the apparatus:

allocating a sequence table header associated with a sequence table in the non-volatile memory prior to writing the sequence table to the non-volatile memory; and

validating the sequence table header after writing the sequence table entries to the non-volatile memory.

**ABSTRACT**

[0052] A method of writing a file to a flash memory is disclosed in which the file includes a number of data fragments and at least one sequence table. The method comprises writing the data fragments to a non-volatile memory; writing the at least one sequence table including sequence table entries which identifies locations of the data fragment written to the non-volatile memory to a volatile memory, and writing the sequence table from the volatile memory to the non-volatile memory if the sequence table is full or if writing the data fragments of the file is completed.



1/5

100

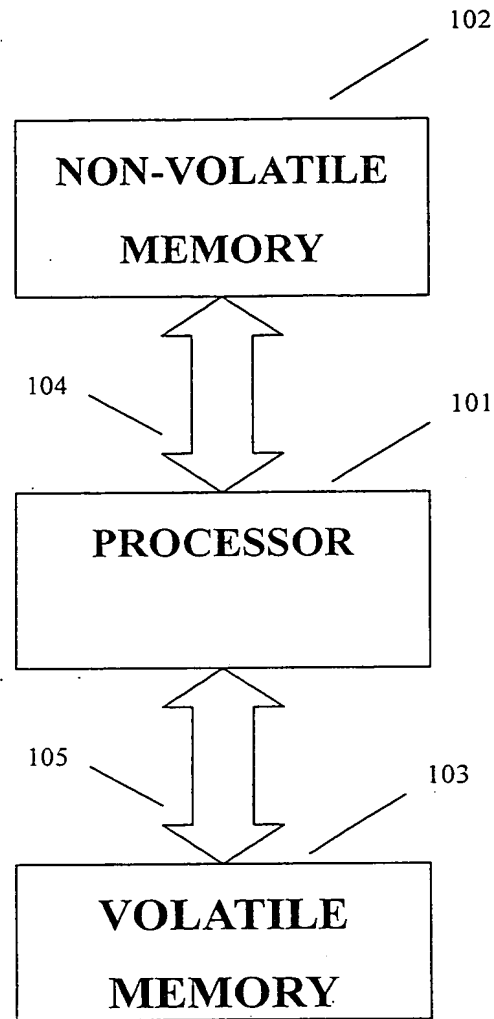


FIG. 1

2/5



ST 221

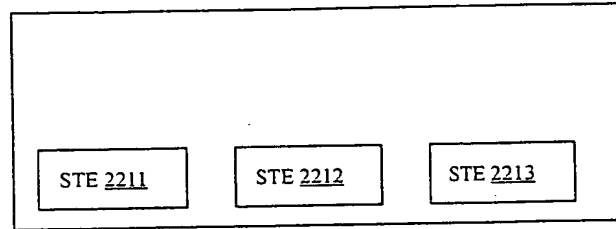


FIG. 3

TI 215

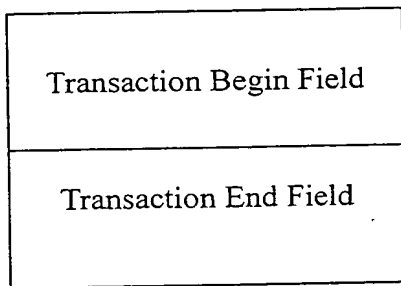


FIG. 4a

STE 2211

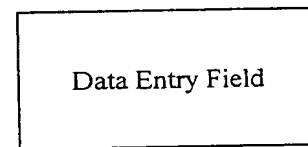


FIG. 4b

STH 221'

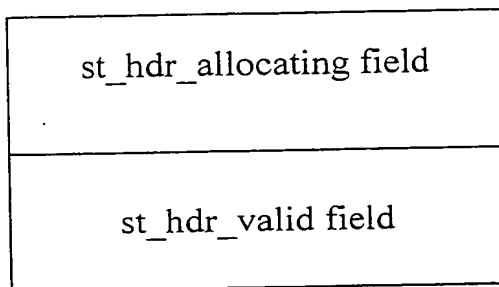


FIG. 4c

DFH 261'

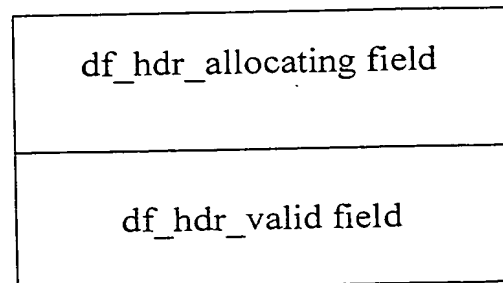


FIG. 4d

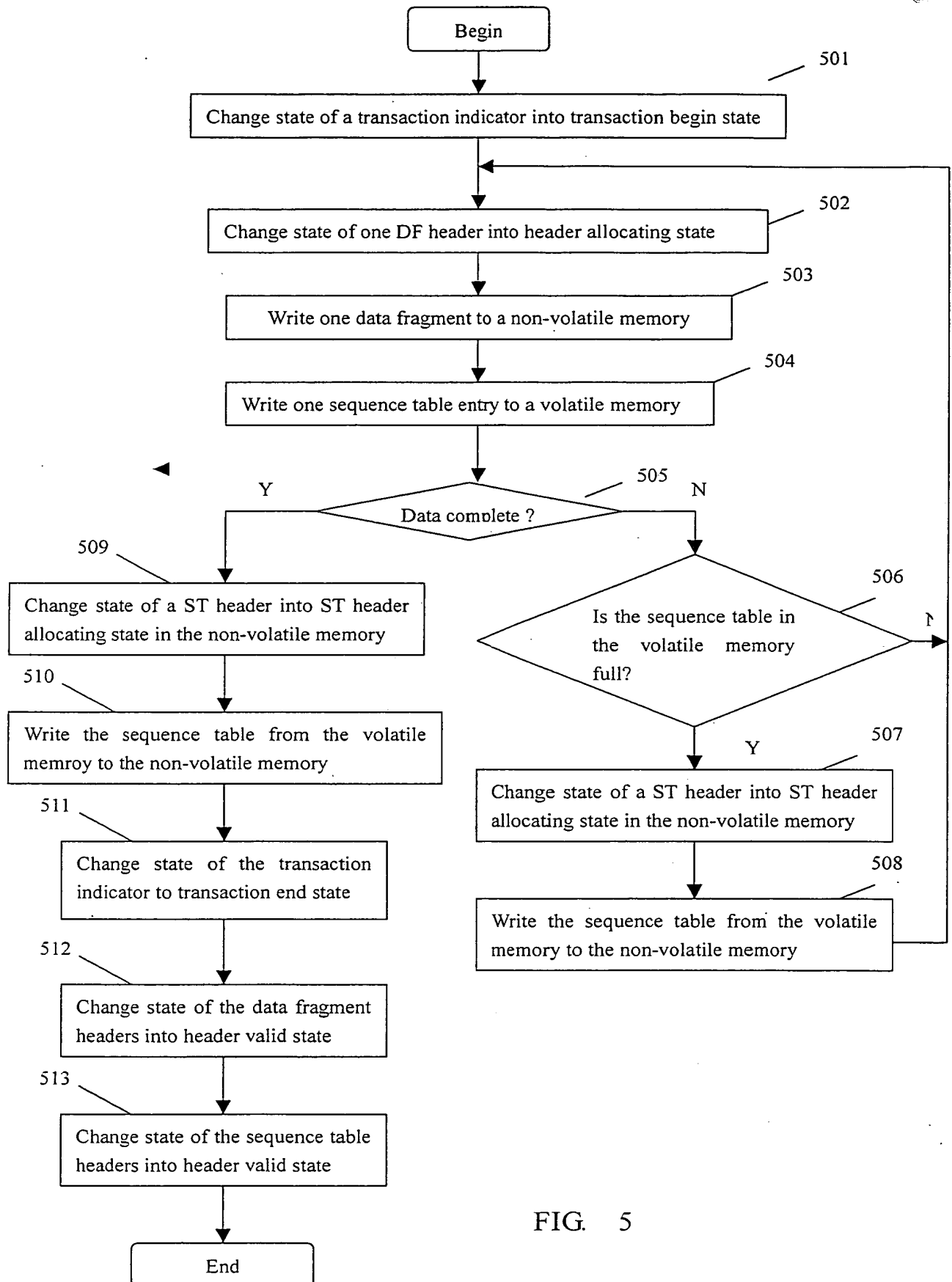


FIG. 5

4/5

Sequence Table	Data Fragment	Write Mode
1	hdr_allocating (1)	1 Bit Twiddling
2	Data Fragment (1)	1 KB Buffer Programming
3	hdr_allocating	1 Bit Twiddling
4	entry_allocating (1)	1 Bit Twiddling
5	data entry (1)	2 Words Programming
6	entry_valid (1)	1 Bit Twiddling
7	hdr_valid (1)	1 Bit Twiddling
8
9	hdr_allocating (64)	1 Bit Twiddling
10	Data Fragment (64)	1 KB Buffer Programming
11	entry_allocating (64)	1 Bit Twiddling
12	data entry (64)	2 Words Programming
13	entry_valid (64)	1 Bit Twiddling
14	hdr_valid	1 Bit Twiddling
15	hdr_valid (64)	1 Bit Twiddling

Bit twiddling (bits)	258
word program (words)	128
Buffer program (bytes)	64K

Sequence Table	Data Fragment	Write Mode
1	transaction begin	1 Bit Twiddling
2	hdr_allocating (1)	1 Bit Twiddling
3	Data Fragment (1)	1 KB Buffer Programming
4
5	hdr_allocating (64)	1 Bit Twiddling
6	Data Fragment (64)	1 KB Buffer Programming
7	hdr_allocating	1 Bit Twiddling
8	data entries	0.5KB Buffer Programming
9	transaction end	1 Bit Twiddling
10	hdr_valid (1)	1 Bit Twiddling
11
12	hdr_valid (64)	1 Bit Twiddling
13	hdr_valid	1 Bit Twiddling
14		

Bit twiddling (bits)	132
word program (words)	0
Buffer program (bytes)	64.5K

Fig. 6

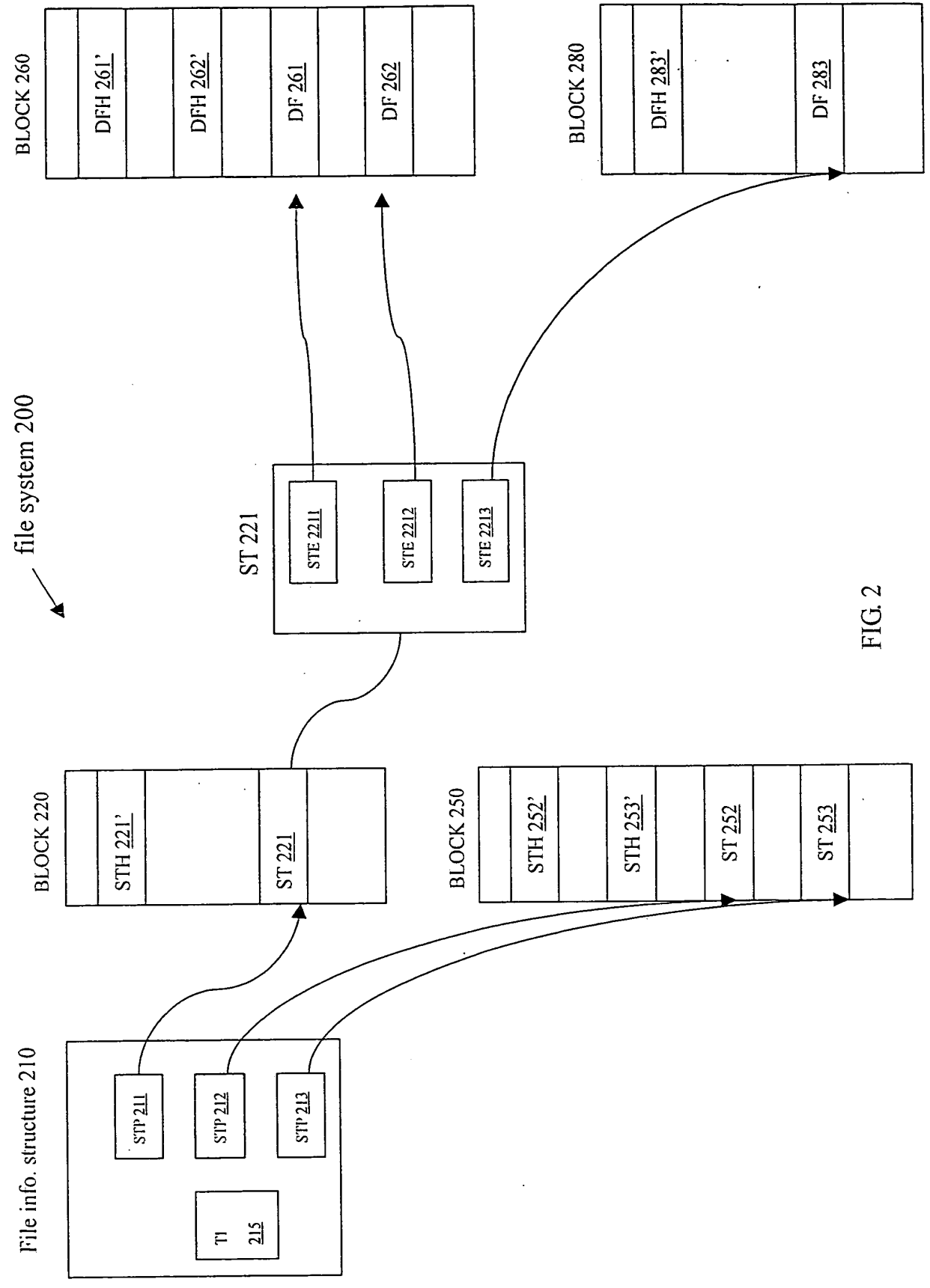


FIG. 2